# Run PySpark Word Count example on Google Cloud Platform using Dataproc

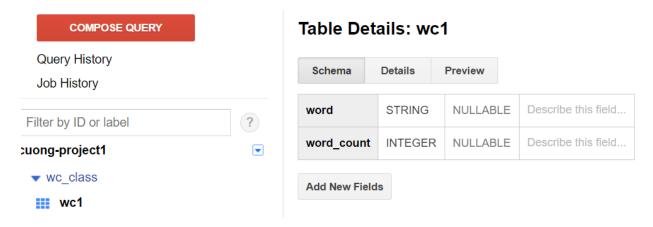
### Overview

This word count example is similar to the one introduced earlier. It will use the Shakespeare dataset in BigQuery. The only difference is that instead of using Hadoop, it uses PySpark which is a Python library for Spark.

### Step 1: create the output table in BigQuery

We need a table to store the output of our Map Reduce procedure.

- Select your project or create a new one, remember to enable billing
- Go to Big Query
- Create a dataset, then a table using the following schema



Example: my project is "cuong-project1", dataset is "wc\_class", and output table is "wc1". The output table has two fields: word and word\_count. We will need these fields to store the output.

### Step 2: create a cluster in Dataproc and Google Cloud Storage

Go to Dataproc and create a cluster similar to our previous tutorial. It should look like this

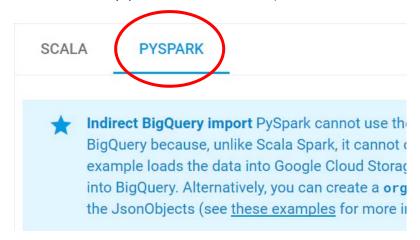


Now click on the link in the "Cloud Storage staging bucket" column, it will bring you to your Cloud Storage. This is where we will upload our python code, which will then give us a link to submit a job to the cluster.

## Step 3: modify the code and upload it to Cloud Storage

You can download the python code 'sparkwc.py' from our piazza website.

You can look at the code here <a href="https://cloud.google.com/dataproc/examples/bigquery-connector-spark-example">https://cloud.google.com/dataproc/examples/bigquery-connector-spark-example</a> (remember to click on the PySpark tab instead of Scala)



```
input_directory =
'gs://{}/hadoop/tmp/bigquery/pyspark_input'.format(bucket)
```

Replace the {} symbols with your Cloud Storage Bucket id.

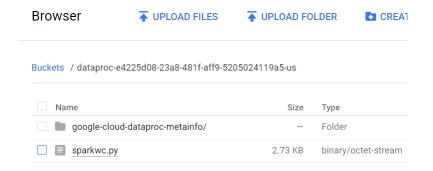
```
#output_dataset = 'wordcount_dataset'
#output_table = 'wordcount_table'
```

Set the corresponding dataset name and table name of your output table (created in Step 1).

```
output_directory =
```

similar to input\_directory, assign the correct Cloud Storage Bucket id here.

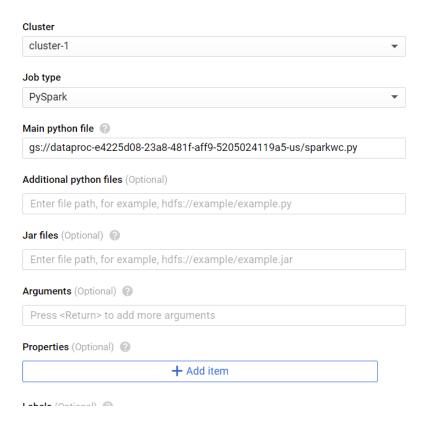
Then, upload the python file to your Cloud Storage and get the link to it. We will use this link as input to our PySpark job.



For example, here I uploaded sparkwc.py, and my link would be gs://dataproc-e4225d08-23a8-481f-aff9-5205024119a5-us/sparkwc.py

# Step 4: submit the job

Similar to our Hadoop example. Use the settings below. Remember to use your own gs:// link, not mine.



Step 5: browse the result

Once the job is done (should be around 2~10 minutes). Go back to Big Query and explore the result.

